



**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re application of: Miller, *et al.* Docket No.: RO999-057  
Serial No.: 09/421,585 Group Art Unit: 2126  
Filed: 10/20/99 Examiner: OPIE, GEORGE L.  
For: APPARATUS AND METHOD FOR PASSIVELY MONITORING LIVENESS OF  
JOBS IN A CLUSTERED COMPUTING ENVIRONMENT

**APPEAL BRIEF**

Mail Stop APPEAL BRIEF - PATENTS  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

Dear Sir/Madam:

This appeal is taken from the Examiner's final rejection, set forth in the Office Action dated 02/25/04, of appellant's claims 1-26. Appellant's Notice of Appeal under 37 C.F.R. § 1.191 was mailed on 05/25/04.

**REAL PARTY IN INTEREST**

International Business Machines Corporation is the Real Party in Interest.

**RELATED APPEALS AND INTERFERENCES**

There are no related appeals or interferences for this patent application.

### **STATUS OF CLAIMS**

As filed, this case included claims 1-26. In response to the first office action dated 09/03/03, an Amendment was filed on 12/02/03 that amended claim 7. In the final rejection, claims 1-26 were rejected under 35 U.S.C. §102(e) as being anticipated by U.S. Patent No. 6,108,699 to Moiin. The claims remaining in the case are claims 1-26, all of which stand finally rejected. No claim has been allowed.

### **STATUS OF AMENDMENTS**

The amendment filed on 12/02/03 has been entered. Therefore, the claims at issue in this appeal are the claims as amended by the amendment filed 12/02/03.

### **SUMMARY OF INVENTION**

An apparatus and method passively determine when a job in a clustered computing environment is dead. Each node in the cluster has a cluster engine for communicating between jobs within the same group on other nodes. A protocol is defined that includes one or more acknowledge (ACK) rounds, and that only performs local processing between ACK rounds. The protocol is executed by jobs that are members of a defined group. Each job in the group has one or more work threads that execute the protocol. In addition, each job has a main thread that communicates between the job and jobs on other nodes (through the cluster engine), routes appropriate messages from the cluster engine to a work thread, and signals to the cluster engine when a fault occurs when the work thread executes the protocol. By assuring that a dead job is reported to other members of the group, liveness information for group members can be monitored without the overhead associated with active liveness checking.

## **ISSUE**

The following single issue is presented for review on this Appeal:

- 1. Whether claims 1-26 are anticipated under 35 U.S.C. §102(e) by Moiin**

## **GROUPING OF CLAIMS**

Claims 1 and 16-18 are grouped, and stand and fall together based on claim 1. Claims 2 and 19 are grouped, and stand and fall together based on claim 2. Claims 3 and 20 are grouped, and stand and fall together based on claim 3. Claims 4 and 21 are grouped, and stand and fall together based on claim 4. Claims 5 and 22 are grouped, and stand and fall together based on claim 5. Claims 6 and 23 are grouped, and stand and fall together based on claim 6. Claims 24-26 are grouped, and stand and fall together based on claim 24. It is appellant's intention that the patentability, *vel non*, of claims 7-15 be considered independently, as these claims do not stand or fall with any other claim. This grouping of claims is appropriate because each set of claims includes a unique combination of limitations not found in the other sets of claims, and because the claims that are not grouped with any other claim include a unique combination of limitations not found in any of the groups.

## ARGUMENT

**Issue 1:        Whether claims 1-26 are anticipated under 35 U.S.C. §102(e)  
by Moiin**

### Claim 1

In the final Office Action dated 02/25/04, the Examiner rejected claims 1-26 under 35 U.S.C. §102(e) as being anticipated by Moiin. In rejecting claim 1, the Examiner reads “applications which execute in any member node of current cluster” in Moiin on the job recited in the claim. A very cursory examination of the claim and of the teachings of Moiin show this mapping does not make sense. A job in claim 1 includes at least one work thread and a main thread. The applications in Moiin may be executed by threads, but do not “include” threads. A job in claim 1 is a process that includes multiple threads, while the applications in Moiin are software applications, which cannot *include* threads. For this reason, the applications in Moiin do not read on the job in claim 1.

The Examiner then reads the main thread 1002 in Moiin on the main thread in claim 1 by stating: “(main thread 1002 ... monitors messages received)”. This is a false statement. The Examiner has quoted language from Moiin in a way that is misleading. Moiin states:

**Main thread 1002** processes state changes of CMM 220A as represented in fields 302-312 in the manner described above and coordinates processing of the other threads of CMM 220A. Sender threads 1004 and receiver threads 1006 send and receive, respectively, reconfiguration messages which indicate to CMM 220A which of the other nodes are operative and in communication with CMM 220A and reconfiguration messages to coordinate reconfiguration of the current cluster. Command reader thread 1008 acts as a remote procedure calling (RPC) server for applications which execute in any member node of the current cluster of which CMM 220A is a member. Transition thread 1010 processes

reconfiguration of the current cluster in accordance with logic flow diagrams 400 (FIG. 4), 500 (FIG. 5), and 600 (FIG. 6) and uses sender threads 1004 and receiver threads 1006 to send and receive, respectively, reconfiguration messages in the manner described above. Communication timeout thread 1012 **monitors messages received** by receiver threads 1002 and detects failure of or, equivalently, loss of communication with a member node of the current cluster.

The bold portions of the citation above show the way the Examiner cited the language. The large portion of text between the bold portions show what is represented by the "...” in the rejection. This shows conclusively that the Examiner has mis-quoted Moiin. Moiin teaches a communication timeout thread 1012 that monitors messages. The main thread 1002 does not monitor messages. For this reason, the main thread 1002 of Moiin does not read on the main thread in claim 1.

Next, the Examiner reads the remote procedure calling of Moiin on the main thread that "routes appropriate messages from the at least one computer system to the at least one work thread". A remote procedure call as taught in Moiin is a way for one computer system to invoke a function (or procedure) in a different computer system. The claim language, in contrast, recites "a main thread . . . that routes appropriate messages from the at least one computer system to the at least one work thread". Note the "at least one work thread" is recited as part of the job at lines 6-7 of claim 1. We see from the express language in claim 1 that the main thread routes messages from at least one computer system coupled to the apparatus to the at least one work thread in the job. The main thread thus performs a routing function of messages from other computer systems to the at least one work thread. The remote procedure call in Moiin is a way for one computer system to invoke functions on a different computer system, and has absolutely nothing to do with the routing of messages from the main thread to one or more work threads within the same computer system. For this reason, the remote procedure call in Moiin does not read on the routing of messages from one computer system to at least one work thread by a main thread.

The Examiner then recites “that signals to the cluster engine when at least one fault occurs when the at least one work thread performs the at least one predefined task”, citing “messages which indicate to CMM 220A which of the other nodes are operative” in Moiin. Again, the Examiner has read a communication between nodes in Moiin on the function recited in claim 1 that occurs between threads on a single node. In claim 1, the main thread signals to the cluster engine in the main thread’s node when at least one fault occurs when the at least one work thread performs the at least one predefined task. This communication all happens on the same node. The main thread thus monitors the progress of the at least one work thread, and signals to the cluster engine when a fault occurs. The passing of messages between nodes to indicate which nodes are operative has nothing to do with this function recited in claim 1. Because Moiin does not teach or suggest a main thread that signals to the cluster engine when at least one fault occurs when the at least one work thread performs the at least one predefined task, claim 1 is allowable over Moiin.

For the many reasons given above, appellant respectfully asserts that claim 1 is allowable over Moiin, and respectfully requests that the Examiner’s rejection of claim 1 under 35 U.S.C. §102(e) be reversed.

#### Claims 16-18

Claims 16-18 are grouped with claim 1, and stand and fall according to the allowability of claim 1.

#### Claim 2

The arguments above with respect to claim 1 apply equally to claim 2, and are incorporated in this section by reference. In rejecting claim 2, the Examiner states: “As to claim 2, Moiin (p6 12-30) teaches a protocol that includes at least one acknowledge

round (each member node of a cluster responds to a reconfiguration message . . . by broadcasting a responding reconfiguration message).” Appellant admits that Moiin teaches a sort of communication protocol where each member node of a cluster responds to a reconfiguration message by broadcasting a responding reconfiguration message. Note, however, that claim 2 expressly recites “wherein the at least one predefined task comprises a protocol that includes at least one acknowledge (ACK) round, and that performs only local processing between ACK rounds.” The Examiner’s rejection of claim 2 fails for two reasons. First, the communication protocol cited by the Examiner in Moiin in rejecting claim 2 is not defined by the at least one predefined task, as recited in claim 2. Second, the Examiner has utterly failed to address the limitation “and that performs only local processing between ACK rounds” in claim 2. As a result, the Examiner has failed to establish a prima facie case of anticipation for claim 2 under 35 U.S.C. §102(e). Claim 2 is directed to the specifics of the at least one predefined task. By including at least one ACK round and performing only local processing between ACK rounds, the invention in claim 2 assures that a work thread cannot get stuck between ACK rounds. Moiin has no such teaching or suggestion. Appellant respectfully asserts that claim 2 is allowable over Moiin. In addition, claim 2 depends on claim 1, which is allowable for the reasons given above. As a result, claim 2 is allowable as depending on an allowable independent claim. Appellant respectfully requests that the Examiner’s rejection of claim 2 under 35 U.S.C. §102(e) be reversed.

#### Claim 19

Claim 19 is grouped with claim 2, and stands and falls according to the allowability of claim 2.

### Claims 3-4

The arguments above with respect to claim 1 apply equally to claims 3-4, and are incorporated in this section by reference. In rejecting claims 3-4, the Examiner states: "As to claims 3-4, Moiin (p.13 24-57) teaches the main thread performs only local processing sans waiting for local resources." However, a careful study of Moiin reveals that the language in Moiin cited by the Examiner has no teaching or suggestion that reads on the limitations in claims 3-4. The Examiner has not mapped any of the teachings of Moiin on the limitations in claims 3-4. Because the Examiner has not specifically identified what teachings in Moiin allegedly read on the limitations in claims 3-4, the Examiner has failed to establish a prima facie case of anticipation for claims 3-4 under 35 U.S.C. §102(e). Nowhere does Moiin teach or suggest that the main thread performs only local processing. For this reason, claim 3 is allowable over Moiin. Nowhere does Moiin teach or suggest that the main thread does not wait for any local resource, and thus is guaranteed to receive a message sent by the cluster engine. For this reason, claim 4 is allowable over Moiin. In addition, claims 3-4 depend on claim 1, which is allowable for the reasons given above. As a result, claims 3-4 are allowable as depending on an allowable independent claim. Appellant respectfully requests that the Examiner's rejection of claims 3-4 under 35 U.S.C. §102(e) be reversed.

### Claim 20

Claim 20 is grouped with claim 3, and stands and falls according to the allowability of claim 3.

### Claim 21

Claim 21 is grouped with claim 4, and stands and falls according to the allowability of claim 4.



### Claims 5-6

The arguments above with respect to claim 1 apply equally to claims 5-6, and are incorporated in this section by reference. In rejecting claims 5-6, the Examiner cites to Moiin at p9 line 51 to p10 line 7. This language in Moiin discusses one or more nodes leaving the cluster, but does not teach the specifics of how this is done. Claim 5 recites an unregistration with the cluster engine. Nowhere does Moiin teach the registration or unregistration with the cluster engine as a way to join or leave a cluster. Claim 6 recites that the unregistration with the cluster engine causes the cluster engine to generate a membership change message. Nowhere does Moiin teach the generation of a membership change message in response to an unregistration with the cluster engine. Because Moiin does not teach the specifics of how nodes leave a cluster, Moiin cannot teach the specific mechanisms in claims 5-6. Appellant respectfully asserts that claims 5-6 are allowable over Moiin. In addition, claims 5-6 depend on claim 1, which is allowable for the reasons given above. As a result, claims 5-6 are allowable as depending on an allowable independent claim. Appellant respectfully requests that the Examiner's rejection of claims 5-6 under 35 U.S.C. §102(e) be reversed.

### Claim 22

Claim 22 is grouped with claim 5, and stands and falls according to the allowability of claim 5.

### Claim 23

Claim 23 is grouped with claim 6, and stands and falls according to the allowability of claim 6.

### Claim 7

The arguments above with respect to claims 1-2 and 5-6 apply equally to claim 7, and are incorporated in this section by reference. In rejecting claim 7, the Examiner states: "As to claim 7, see the discussions of claims 1-2 and 6 supra. The limitations in claim 7 are an amalgamation of the features recited in claims 1-2 and 6. Hence, Moiin's teachings regarding claims 1-2 and 6 are similarly applied to the corresponding claim 7 limitations." Appellant respectfully asserts that the Examiner has failed to establish a prima facie case of anticipation under 35 U.S.C. §102(e) for claim 7.

Claim 7 recites limitations that are not found in any of claims 1-2 or 6. Claim 7 recites a cluster of computer systems that each includes the limitations in lines 3-21 of claim 7. None of claims 1-2 or 6 contain the limitation of such a cluster. In addition, claim 7 recites a network interface at lines 3-4 that is not addressed in any of claims 1-2 or 6. As a result, the Examiner has failed to establish a prima facie case of anticipation for claim 7 under 35 U.S.C. §102(e). In addition, claim 7 contains many of the limitations addressed above in claims 1-2 and 5-6 that patentably distinguish over Moiin. As a result, claim 7 is allowable over Moiin for the same reasons given above with respect to claims 1-2 and 5-6. Appellant respectfully requests that the Examiner's rejection of claim 7 under 35 U.S.C. §102(e) be reversed.

### Claim 8

The arguments above with respect to claims 1 and 7 apply equally to claim 8, and are incorporated in this section by reference. The Examiner summarily rejects all of claims 8-14 by stating:

As to claims 8-14, note the rejections of claims 1-7 above. Claims 8-14 are the same as claims 1-7, except claims 8-14 are method claims and claims 1-7 are apparatus claims.

While claims 8-14 include many limitations that are also in claims 1-7, this fact does not excuse the Examiner from addressing all of the limitations in claims 8-14. For example, claim 8 recites the steps of “defining a task.” The Examiner has not indicated what teachings in Moiin read on this method step. Claim 8 also recites “providing at least one work thread for each job that executes the task.” Nowhere does Moiin teach providing a work thread for each job that executes a task. Because the Examiner has failed to address the specific method claim steps recited in claims 8-14, the Examiner has failed to establish a prima facie case of anticipation for claims 8-14 under 35 U.S.C. §102(e). Claim 8 does include many limitations that are common to claims 1-7, and is therefore allowable for the reasons given above with respect to claims 1-7. Appellant respectfully requests that the Examiner’s rejection of claim 8 under 35 U.S.C. §102(e) be reversed.

#### Claim 9

The arguments above with respect to claims 1, 2, 7 and 8 apply equally to claim 9, and are incorporated in this section by reference. Claim 9 depends on claim 8. Because the Examiner failed to establish a prima facie case of anticipation for claim 8 by failing to address all of the limitations in claim 8, the Examiner has likewise failed to establish a prima facie case of anticipation for claim 9 because claim 9 depends on claim 8. In addition, claim 9 includes limitations similar to those found in claim 2, and is therefore allowable for the reasons given above with respect to claim 2. Appellant respectfully requests that the Examiner’s rejection of claim 9 under 35 U.S.C. §102(e) be reversed.

#### Claim 10

The arguments above with respect to claims 1, 3, 7 and 8 apply equally to claim 10, and are incorporated in this section by reference. Claim 10 depends on claim 8. Because the Examiner failed to establish a prima facie case of anticipation for claim 8 by

failing to address all of the limitations in claim 8, the Examiner has likewise failed to establish a prima facie case of anticipation for claim 10 because claim 10 depends on claim 8. In addition, claim 10 includes limitations similar to those found in claim 3, and is therefore allowable for the reasons given above with respect to claim 3. Appellant respectfully requests that the Examiner's rejection of claim 10 under 35 U.S.C. §102(e) be reversed.

#### Claim 11

The arguments above with respect to claims 1, 4, 7 and 8 apply equally to claim 11, and are incorporated in this section by reference. Claim 11 depends on claim 8. Because the Examiner failed to establish a prima facie case of anticipation for claim 8 by failing to address all of the limitations in claim 8, the Examiner has likewise failed to establish a prima facie case of anticipation for claim 11 because claim 11 depends on claim 8. In addition, claim 11 includes limitations similar to those found in claim 4, and is therefore allowable for the reasons given above with respect to claim 4. Appellant respectfully requests that the Examiner's rejection of claim 11 under 35 U.S.C. §102(e) be reversed.

#### Claim 12

The arguments above with respect to claims 1, 5, 7 and 8 apply equally to claim 12, and are incorporated in this section by reference. Claim 12 depends on claim 8. Because the Examiner failed to establish a prima facie case of anticipation for claim 8 by failing to address all of the limitations in claim 8, the Examiner has likewise failed to establish a prima facie case of anticipation for claim 12 because claim 12 depends on claim 8. In addition, claim 12 includes limitations similar to those found in claim 5, and is therefore allowable for the reasons given above with respect to claim 5. Appellant

respectfully requests that the Examiner's rejection of claim 12 under 35 U.S.C. §102(e) be reversed.

### Claim 13

The arguments above with respect to claims 1, and 6-8 apply equally to claim 13, and are incorporated in this section by reference. Claim 13 depends on claim 12, which depends on claim 8. Because the Examiner failed to establish a prima facie case of anticipation for claim 8 by failing to address all of the limitations in claim 8, the Examiner has likewise failed to establish a prima facie case of anticipation for claim 13 because claim 13 depends on claim 8. In addition, claim 13 includes limitations similar to those found in claim 6, and is therefore allowable for the reasons given above with respect to claim 6. Appellant respectfully requests that the Examiner's rejection of claim 13 under 35 U.S.C. §102(e) be reversed.

### Claim 14

The arguments above with respect to claims 1-8 apply equally to claim 14, and are incorporated in this section by reference. Claim 14 includes many method steps that are not taught in the apparatus claims. For example, claim 14 includes the limitation at lines 4-5 of:

defining a protocol that includes at least one acknowledge (ACK) round, and that performs only local processing between ACK rounds;

The Examiner's rejection of claim 14 relies upon the rejection of claim 7. However, the rejection of claim 7 does not address this specific method step. Claim 14 also recites:

wherein the main thread performs only local processing and does not wait for any local resource, and thus is guaranteed to receive a message sent by the cluster engine;

The Examiner's rejection of claim 14 relies upon the rejection of claim 7. However, the rejection of claim 7 does not address these limitations. Claim 14 also recites:

unregistering with the cluster engine when at least one fault occurs during execution of the protocol.

The Examiner's rejection of claims 1-7 does not address this specific method step. We see from these three simple examples above that the Examiner has not addressed all of the limitations in claim 14, and has therefore failed to establish a prima facie case of anticipation for claim 14 under 35 U.S.C. §102(e).

The rejection of claims 5-6 vaguely addresses reconfiguration of a cluster in Moiin "to form a new cluster which does not include any failed nodes." However, Moiin does not teach unregistering with the cluster engine *when at least one fault occurs during the execution of the protocol*, as recited in claim 14. For this reason, claim 14 is allowable over Moiin. In addition, there are many limitations in claim 14 that are similar to those recited in claims 1-13. As a result, claim 14 is allowable for the reasons given above with respect to claims 1-13. Appellant respectfully requests that the Examiner's rejection of claim 14 under 35 U.S.C. §102(e) be reversed.

#### Claim 15

In rejecting claim 15, the Examiner relies upon the rejection of claim 6. Note, however, that claim 15 includes the limitations of claim 14 upon which it depends, and the limitations in claim 14 were not addressed by the Examiner. As a result, the Examiner has failed to establish a prima facie case of anticipation for claim 15 under 35 U.S.C. §102(e). Appellant respectfully asserts that Moiin does not teach the unique combination of features in claim 15, and requests that the Examiner's rejection of claim 15 under 35 U.S.C. §102(e) be reversed.

#### Claim 24

In rejecting claim 24, the Examiner states that claim 24 is the same as claim 1, except claim 24 is a computer program product claim and claim 1 is an apparatus claim. This statement is incorrect. Claim 24 contains limitations not found in claim 1. For example, claim 24 recites that the protocol only performs local tasks between ACK rounds. Claim 24 also recites unregistering with the cluster engine, which causes the cluster engine to generate a membership change to remaining members of the group. Because these limitations are not found in claim 1, the Examiner's reliance upon claim 1 in rejecting claim 24 results in the Examiner failing to establish a prima facie case of anticipation for claim 24 under 35 U.S.C. §102(e). Claim 24 contains many of the limitations addressed above in claims 1-2 and 5-6 that patentably distinguish over Moiin. As a result, claim 24 is allowable over Moiin for the same reasons given above with respect to claims 1-2 and 5-6. Appellant respectfully requests that the Examiner's rejection of claim 24 under 35 U.S.C. §102(e) be reversed.

#### Claims 25-26

Claims 25 and 26 are grouped with claim 24, and stand and fall according to the allowability of claim 24.

#### Response to Arguments in Final Office Action

In the Response to Applicant's Arguments section at page 4 of the final office action, the Examiner states:

Applicant argues (claims 1-26) that Moiin's teachings do not read-on [sic] the process signaling for computing cluster management as claimed. Contrary to Applicant's contention, the Moiin reference does teach the recited acknowledge signaling for cluster membership maintenance.

This language by the Examiner provides some insight into the Examiner's thought processes. Applicant never argued that Moiin's teachings do not read on "the process signaling for computing cluster management" as claimed. This language is not in the claims, and has not been advanced by appellant. This language is the Examiner's own interpretation of the claim language, rather than addressing the specific language used in the claims. This shows that the Examiner is taking an approach to examining the claims that is broader than the express claim language allows. In the last two sentences of the Response to Applicant's Arguments, the Examiner also states:

The fact that Applicant has not narrowed the definition/scope of the current claims implies that Applicant intends an extensive coverage breadth of the claims, which is met by the cited prior art. Consequently, the teachings of Moiin do read-on [sic] the cluster control using multithreaded systems to process membership messages, as presented in the pending claims.

Wow. This first sentence by the Examiner is amazing. The Examiner seems to impute some breadth to the claims beyond their literal scope simply because they have not been amended. This shows the Examiner's misunderstanding of the proper procedures for examining claims. Appellant intends claim coverage that is commensurate with their literal scope. Of course, these claims may also be imputed a scope in the future that is broader than their literal scope by appropriate application of the Doctrine of Equivalents. All the appellant asks is that the Examiner consider each and every limitation that is expressly recited in the claims. The second sentence quoted above is even more revealing. The Examiner states that the teachings of Moiin read on "the cluster control using multithreaded systems to process membership messages, as presented in the pending claims." Now we get to the nut of the problem. The Examiner is taking a high-level view of the claims, utterly ignoring many of the detailed limitations in the claims. Appellant admits that the claims and Moiin both teach clustered computing, both teach multi-threading, and both teach membership changes to the cluster. This is not the appropriate inquiry. The question is whether Moiin teaches all of the limitations



expressly recited in the claims. This high-level view of the Examiner ignores many of the details that are expressly recited in the claims. His summary quoted above shows beyond doubt that he is interpreting the claim language in a manner that is broader than the express teachings in the claims. While the Examiner has a duty to interpret the claims as broadly as reasonably possible during prosecution, the Examiner CANNOT under ANY CIRCUMSTANCES simply ignore express limitations in the claims. By taking a high-level view of the claims, the Examiner has lost sight of the details expressly recited in the claims. Appellant respectfully asserts that the Examiner's broad and liberal reading of Moiin stretches Moiin clearly beyond its reasonable bounds. Moiin does not teach many of the detailed limitations in the claims.

In claim 1, which is representative of the broadest claims, there are many limitations that are not taught in Moiin. Moiin does not teach or suggest a job that includes at least one work thread and a main thread. Moiin does not teach or suggest that the work thread performs at least one predefined task. Moiin does not teach or suggest that the main thread receives messages, routes messages to the work thread(s), and signals to the cluster engine when a fault occurs when the work thread performs the predefined task. As a result, the claims are all allowable over Moiin for the many reasons discussed in detail above.

It is interesting to note that most of the arguments presented herein were also presented in the Amendment filed on 12/02/03. In the Response to Applicant's Arguments section of the final office action, the Examiner has not addressed the vast majority of these arguments. Hopefully the Examiner will do a better job in the Answer in addressing each and every one of the pending claims and appellant's arguments in support thereof.

### General Comments

The Examiner's claim rejections are based upon a text printout of the Moiin patent, citing page numbers and line numbers. This second version of Moiin makes it impossible to determine from a standard printed patent what portions of Moiin the Examiner is referring to when citing page numbers and line numbers of the text printout. Because the line numbers in the text printout are not numbered, the Examiner's reference to page numbers and line numbers of the text printout forces the reader to manually count lines to determine which portions of Moiin the Examiner is referring to. This is a silly exercise given the clearly-marked column and line numbers in the printed patent.

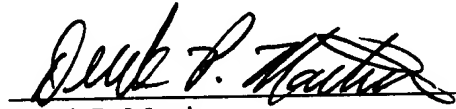
Appellant requests the Board to instruct the Examiner to use the standard column and line number references to the printed patent in future office actions. This will allow anyone who needs to review the office action to refer to the standard printed patent, rather than having to retrieve the text version in the file wrapper.

## **CONCLUSION**

Claims 1-26 are addressed in this Appeal. For the numerous reasons articulated above, appellant maintains that the rejection of claims 1-26 under 35 U.S.C. § 102(e) is erroneous.

Appellant respectfully submits that this Appeal Brief fully responds to, and successfully contravenes, every ground of rejection and respectfully requests that the final rejection be reversed and that all claims in the subject patent application be found allowable.

Respectfully submitted,

A handwritten signature in cursive script, appearing to read "Derek P. Martin", written over a horizontal line.

Derek P. Martin  
Reg. No. 36,595

**MARTIN & ASSOCIATES, L.L.C.**  
P.O. Box 548  
Carthage, MO 64836-0548  
(417) 358-4700  
Fax (417) 358-5757

## **APPENDIX - CLAIMS FINALLY REJECTED**

- 1    1. An apparatus comprising:
  - 2            at least one processor;
  - 3            a memory coupled to the at least one processor;
  - 4            a cluster engine residing in the memory and executed by the at least one
  - 5 processor;
  - 6            a job residing in the memory and executed by the at least one processor, the job
  - 7 including:
    - 8                    at least one work thread that performs at least one predefined task; and
    - 9                    a main thread that receives messages from at least one computer system
    - 10                  coupled to the apparatus, that routes appropriate messages from the at least one
    - 11                  computer system to the at least one work thread, and that signals to the cluster
    - 12                  engine when at least one fault occurs when the at least one work thread performs
    - 13                  the at least one predefined task.
- 1    2. The apparatus of claim 1 wherein the at least one predefined task comprises a protocol
- 2    that includes at least one acknowledge (ACK) round, and that performs only local
- 3    processing between ACK rounds.
- 1    3. The apparatus of claim 1 wherein the main thread performs only local processing.
- 1    4. The apparatus of claim 1 wherein the main thread does not wait for any local resource,
- 2    and thus is guaranteed to receive a message sent by the cluster engine.
- 1    5. The apparatus of claim 1 wherein the signal to the cluster engine comprises an
- 2    unregistration with the cluster engine

1 6. The apparatus of claim 5 wherein the unregistration with the cluster engine causes the  
2 cluster engine to generate a membership change message.

1 7. A networked computer system comprising:  
2 a cluster of computer systems that each includes:  
3 a network interface that couples each computer system via a network to  
4 other computer systems in the cluster;  
5 at least one processor;  
6 a memory coupled to the at least one processor;  
7 a cluster engine residing in the memory and executed by the at least one  
8 processor;  
9 a job residing in the memory and executed by the at least one processor,  
10 the job including:  
11 at least one work thread that executes a predefined protocol that  
12 includes at least one acknowledge (ACK) round, wherein the protocol only  
13 performs local tasks between ACK rounds; and  
14 a main thread that registers with the cluster engine to become a  
15 member of a group, that receives messages from at least one computer  
16 system coupled to the apparatus, that routes appropriate messages from the  
17 at least one computer system to the at least one work thread, and that  
18 signals to the cluster engine when at least one fault occurs when the at  
19 least one work thread performs the at least one predefined task by  
20 unregistering with the cluster engine, wherein unregistering with the  
21 cluster engine causes the cluster engine to generate a membership change  
22 to remaining members of the group.

1 8. A computer-implemented method for notifying jobs that form a group in a clustered  
2 computing environment when a member of the group is no longer alive, the method  
3 comprising the steps of:  
4       defining a task;  
5       providing a cluster engine for each member of the group that communicates with  
6 the other cluster engines in the group;  
7       providing at least one work thread for each job that executes the task;  
8       providing a main thread for each job, the main thread performing the steps of:  
9           receiving messages from other members of the group via the cluster engine  
10          corresponding to the main thread;  
11          routing appropriate messages from the other members of the group to the  
12          at least one work thread; and  
13          signaling to the cluster engine when at least one fault occurs during the  
14          execution of the task by the work thread.

1 9. The method of claim 8 wherein the task comprises a protocol that includes at least one  
2 acknowledge (ACK) round, and that performs only local processing between ACK  
3 rounds.

1 10. The method of claim 8 wherein the main thread performs only local processing.

1 11. The method of claim 8 wherein the main thread does not wait for any local resource,  
2 and thus is guaranteed to receive a message sent by the cluster engine.

1 12. The method of claim 8 wherein the step of signaling to the cluster engine comprises  
2 the step of unregistering with the cluster engine.

1 13. The method of claim 12 wherein the step of unregistering with the cluster engine  
2 causes the cluster engine to generate a membership change message to remaining  
3 members of the group.

1 14. A computer-implemented method for notifying jobs that form a group in a clustered  
2 computing environment when a member of the group is no longer alive, the method  
3 comprising the steps of:

4 defining a protocol that includes at least one acknowledge (ACK) round, and that  
5 performs only local processing between ACK rounds;

6 providing a cluster engine for each member of the group that communicates with  
7 the other cluster engines in the group;

8 providing at least one work thread for each job that executes at least a portion of  
9 the protocol;

10 providing a main thread for each job, the main thread performing the steps of:

11 registering with the cluster engine to become a member of the group;

12 receiving messages from other members of the group via the cluster engine  
13 corresponding to the main thread;

14 routing appropriate messages from the other members of the group to the  
15 at least one work thread;

16 wherein the main thread performs only local processing and does not wait  
17 for any local resource, and thus is guaranteed to receive a message sent by the  
18 cluster engine;

19 unregistering with the cluster engine when at least one fault occurs during  
20 execution of the protocol.

1 15. The method of claim 14 wherein the step of unregistering with the cluster engine  
2 causes the cluster engine to generate a membership change message to remaining  
3 members of the group.

1 16. A program product comprising:  
2 (A) a computer program comprising:  
3 at least one work thread that performs at least one predefined task; and  
4 a main thread that receives messages from a corresponding cluster engine,  
5 that routes appropriate messages from the cluster engine to the at least one work  
6 thread, and that signals to the cluster engine when at least one fault occurs when  
7 the at least one work thread performs the at least one predefined task; and  
8 (B) signal bearing media bearing the computer program.

1 17. The program product of claim 16 wherein the signal bearing media comprises  
2 recordable media.

1 18. The program product of claim 16 wherein the signal bearing media comprises  
2 transmission media.

1 19. The program product of claim 16 wherein the at least one predefined task comprises  
2 a protocol that includes at least one acknowledge (ACK) round, and that performs only  
3 local processing between ACK rounds.

1 20. The program product of claim 16 wherein the main thread performs only local  
2 processing.

1 21. The program product of claim 16 wherein the main thread does not wait for any local  
2 resource, and thus is guaranteed to receive a message sent by the cluster engine.

1 22. The program product of claim 16 wherein the signal to the cluster engine comprises  
2 an unregistration with the cluster engine



1 23. The program product of claim 22 wherein the unregistration with the cluster engine  
2 causes the cluster engine to generate a membership change message.

1 24. A program product comprising:

2 (A) a computer program comprising:

3 at least one work thread that performs a predefined protocol that  
4 includes at least one acknowledge (ACK) round, wherein the protocol only  
5 performs local tasks between ACK rounds; and

6 a main thread that registers with the cluster engine to become a  
7 member of a group, that receives messages from at least one computer  
8 system coupled to the apparatus, that routes appropriate messages from the  
9 at least one computer system to the at least one work thread, and that  
10 signals to the cluster engine when at least one fault occurs when the at  
11 least one work thread performs the at least one predefined task by  
12 unregistering with the cluster engine, wherein unregistering with the  
13 cluster engine causes the cluster engine to generate a membership change  
14 to remaining members of the group; and

15 (B) signal bearing media bearing the computer program.

1 25. The program product of claim 24 wherein the signal bearing media comprises  
2 recordable media.

1 26. The program product of claim 24 wherein the signal bearing media comprises  
2 transmission media.